EXHIBIT A

Clean Version of the Specification as Amended:

The paragraph beginning at column (hereafter "col.") 1, line 4:

This application claims the benefit of U.S. Provisional Application No. 60/033,271 for PUBLIC KEY CRYTOGRAPHIC APPARATUS AND METHOD, filed Dec. 9, 1996, naming as inventors, Thomas Collins, Dale Hopkins, Susan Langford and Michael Sabin, the disclosure of which is incorporated by reference.

The paragraph beginning at col. 1, line 64:

The RSA scheme capitalizes on the relative ease of creating a composite number from the product of two prime numbers whereas the attempt to factor the composite number into its constituent primes is difficult. The RSA scheme uses a public key E comprising a pair of positive integers n and e, where n is a composite number of the form

$$n=p\cdot q \qquad (1)$$

where p and q are different prime numbers, and e is a number relatively prime to (p-1) and (q-1); that is, e is relatively prime to (p-1) or (q-1) if e has no factors in common with either of them. Importantly, the sender has access to n and e, but not to p and q. The message M is a number representative of a message to be transmitted wherein

$$0\leq M\leq n-1. \qquad (2)$$

The sender enciphers M to create ciphertext C by computing the exponential

$$C\equiv M^{e}(\mathrm{mod}\ n). \qquad (3)$$

The paragraph beginning at col. 2, line 19:

The recipient of the ciphertext C retrieves the message M using a (private) decoding key D, comprising a pair of positive integers d and n, employing the relation

$$C\equiv M^{d}(\mathrm{mod}\ n) \qquad (4)$$

As used in (4), above, d is a multiplicative inverse of

$$e(\text{mod}(\text{lcm}((p\text{-}1), (q\text{-}1)))) \quad (5)$$

so that

$$e \cdot d \equiv 1(\text{mod}(\text{lcm}((p\text{-}1), (q\text{-}1)))) \quad (6)$$

where lcm((p-1), (q-1)) is the least common multiple of numbers p-1 and q-1. Most commercial implementations of RSA employ a different, although equivalent, relationship for obtaining d:

$$d \equiv e^{-1} \text{ mod}((p\text{-}1)\cdot(q\text{-}1)). \quad (7)$$

This alternate relationship simplifies computer processing.

The paragraph beginning at col. 3, line 23:

It is still another object of this invention to provide a system and method for implementing an RSA scheme in which the factors of n do not increase in length as n increases in length.

The paragraph beginning at col. 3, line 27:

It is still another object to provide a system and method for utilizing multiple (more than two), distinct prime number factors to create n.

The paragraph beginning at col. 3, line 36:

The present invention discloses a method and apparatus for increasing the computational speed of RSA and related public key schemes by focusing on a neglected area of computation inefficiency. Instead of $n=p\cdot q$, as is universal in the prior art, the present invention discloses a method and apparatus wherein n is developed from three or more distinct random prime numbers; i.e., $n=p_1\cdot p_2 \cdot \ldots \cdot p_k$, where k is an integer greater than 2 and p_1, p_2,. . . p_k are sufficiently large distinct random primes. Preferably, "sufficiently large primes" are prime numbers that are numbers approximately 150 digits

long or larger. The advantages of the invention over the prior art should be immediately apparent to those skilled in this art. If, as in the prior art, p and q are each on the order of, say, 150 digits long, then n will be on the order of 300 digits long. However, three primes p_1, p_2 and p_3 employed in accordance with the present invention can each be on the order of 100 digits long and still result in n being 300 digits long. Finding and verifying 3 distinct primes, each 100 digits long, requires significantly fewer computational cycles than finding and verifying 2 primes each 150 digits long.

The paragraph beginning at col. 3, line 56:

The commercial need for longer and longer primes shows no evidence of slowing; already there are projected requirements for n of about 600 digits long to forestall incremental improvements in factoring techniques and the ever faster computers available to break ciphertext. The invention, allowing 4 primes each about 150 digits long to obtain a 600 digit n, instead of two primes about 300 digits long, results in a marked improvement in computer performance. For, not only are primes that are 150 digits in size easier to find and verify than ones on the order of 300 digits, but by applying techniques the inventors derive from the Chinese Remainder Theorem (CRT), public key cryptography calculations for encryption and decryption are completed much faster--even if performed serially on a single processor system. However, the inventors' techniques are particularly adapted to advantageously apply RSA public key cryptographic operations to parallel computer processing.

The paragraph beginning at col. 4, line 6:

The present invention is capable of extending the RSA scheme to perform encryption and decryption operation using a large (many digit) n much faster than heretofore possible. Other advantages of the invention include its employment for decryption without the need to revise the RSA public key encryption transformation scheme currently in use on thousands of large and small computers.

The paragraph beginning at col. 4, line 13:

A key assumption of the present invention is that n, composed of 3 or more sufficiently large distinct prime numbers, is no easier (or not very much easier) to factor than the prior art, two prime number n. The assumption is based on the observation that there is no indication in the prior art literature that it is "easy" to factor a product consisting of more than two sufficiently large, distinct prime numbers. This assumption may be justified given the continued effort (and failure) among experts to find a way "easily" to break large composite numbers into their large prime factors. This assumption is similar, in the inventors' view, to the assumption underlying the entire field of public key cryptography that factoring composite numbers made up of two distinct primes is not "easy." That is, the entire field of public key cryptography is based not on mathematical proof, but on the assumption that the empirical evidence of failed sustained efforts to find a way systematically to solve NP problems in polynomial time indicates that these problems truly are "difficult."

The paragraph beginning at col. 4, line 32:

The invention is preferably implemented in a system that employs parallel operations to perform the encryption, decryption operations required by the RSA scheme. Thus, there is also disclosed a cryptosystem that includes a central processor unit (CPU) coupled to a number of exponentiator elements. The exponentiator elements are special purpose arithmetic units designed and structured to be provided message data M, an encryption key e, and a number n (where $n= p_1 \cdot p_2 \cdot \ldots \cdot p_k$, k being greater than 2) and return ciphertext C according to the relationship,

$$C \equiv M^e \pmod{n}.$$

The paragraph beginning at col. 4, line 45:

Alternatively, the exponentiator elements may be provided the ciphertext C, a decryption (private) key d and n to return M according to the relationship,

$$M \equiv C^d \pmod{n}$$

The paragraph beginning at col. 4, line 50:

According to this decryption aspect of the invention, the CPU receives a task, such as the requirement to decrypt ciphertext data C. The CPU will also be provided, or have available, a [public] private key [e] d and n, and the factors of n (p_1, p_2, . . . p_k). The CPU breaks the decryption task down into a number of sub-tasks, and delivers the sub-tasks to the exponentiator elements. The results of the sub-tasks are returned by the exponentiator elements to the CPU which , using a form of the CRT, combines the results to obtain the message data M. An encryption task may be performed essentially in the same manner by the CPU and its use of the exponentiator elements. However, usually the factors of n are not available to the sender (encryptor), only the public key, e and n, so that no sub-tasks are created.

Before the paragraph beginning at col. 5, line 52 the following new paragraph:

Alternatively, a message data M can be encoded with the private key to a signed message data M_s using a relationship of the form

$$M_s \equiv M^d \pmod{n}.$$

The message data M can be reproduce from the signed message data M_S by decoding the signed data with the public key, using a relationship of the form

$$M \equiv M_s^{\,e} \pmod{n}.$$

The paragraph beginning at col. 5, line 30:

According to the present invention, the public key portion e is picked. Then, three or more random large, distinct prime numbers, p_1, p_2, . . . , p_k are developed and checked to ensure that each (p_i-1) is relatively prime to e. Preferably, the prime numbers are of equal length. Then, the product $n = p_1 \cdot p_2 \cdot \ldots p_k$ is computed.

The paragraph beginning at col. 5, line 36:

Finally, the decryption exponent, d, is established by the relationship:

$$d \equiv e^{-1} \bmod ((p_1 - 1)\cdot(p_2 - 1)\cdot \ldots \cdot(p_k - 1)), \text{ or equivalently}$$

$$d \equiv e^{-1} \bmod (\text{lcm}((p_1 - 1), (p_2 - 1), \ldots (p_k - 1)))$$

The paragraph beginning at col. 5, line 41:

The message data, M is encrypted to ciphertext C using the relationship of (3), above, i.e.,

$$C \equiv M^e \pmod{n}$$

The paragraph beginning at col. 5, line 46:

To decrypt the ciphertext, C, the relationship of (4), above, is used:

$$M \equiv C^d \pmod{n}$$

where n and d are those values identified above.

The paragraph beginning at col. 5, line 52:

Using the present invention involving three primes to develop the product n, RSA encryption and decryption time can be substantially less than an RSA scheme using two primes by dividing the encryption or decryption task into sub-tasks, one sub-task for each distinct prime. (However, breaking the encryption or decryption into subtasks requires knowledge of the factors of n. This knowledge is not usually available to anyone except the owner of the key, so the encryption process can be accelerated only in special cases, such as encryption for local storage. A system encrypting data for another user performs the encryption process according to (3), independent of the number of factors of n. Decryption, on the other hand, is performed by the owner of a key, so the factors of n are

generally known and can be used to accelerate the process.) For example, assume that three distinct primes, p_1, p_2, and p_3, are used to develop the product n. Thus, decryption of the ciphertext, C, using the relationship

$$M \equiv C^d \pmod{n}$$

is used to develop the decryption sub-tasks:

$$M_1 \equiv C_1^{d_1} \pmod{p_1}$$

$$M_2 \equiv C_2^{d_2} \pmod{p_2}$$

$$M_3 \equiv C_3^{d_3} \pmod{p_3}$$

where

$$C_1 \equiv C \pmod{p_1};$$

$$C_2 \equiv C \pmod{p_2};$$

$$C_3 \equiv \mathrm{C} \pmod{p_3};$$

$$d_1 \equiv d \pmod{(p_1 - 1)};$$

$$d_2 \equiv d \pmod{(p_2 - 1)}; \text{ and}$$

$$d_3 \equiv d \pmod{(p_3 - 1)}.$$

The paragraph beginning at col. 6, line 24:

The results of each sub-task, M_1, M_2, and M_3 can be combined to produce the plaintext, M, by a number of techniques. However, it is found that they can most expeditiously be combined by a form of the Chinese Remainder Theorem (CRT) using, preferably, a recursive scheme. Generally, the plaintext M is obtained from the combination of the individual sub-tasks by the following relationship:

$$Y_i \equiv Y_{i-1} + ((M_i - Y_{i-1}) (w_i^{-1} \pmod{p_i})) \pmod{p_i}) \cdot w_i \pmod{n}$$

where $2 \leq i \leq k$ where k is the number of prime factors of n, and

$$M=Y_k,\ Y_1=C_1,\ and\ w_i=\prod_{j<i} p_j$$

Encryption is performed in much the same manner as that used to obtain the plaintext M, provided (as noted above) the factors of n are available. Thus, the relationship

$$C \equiv M^e \pmod{n},$$

can be broken down into the three sub-tasks,

$$C_1 = M_1^{e_1} \pmod{p_1},$$

$$C_2 = M_2^{e_2} \pmod{p_2} \quad \text{and}$$

$$C_3 = M_3^{e_3} \pmod{p_3}.$$

where

$$M_1 \equiv M \pmod{p_1},$$

$$M_2 \equiv M \pmod{p_2},$$

$$M_3 \equiv M \pmod{p_3},$$

$$e_1 \equiv e \bmod (p_1 - 1),$$

$$e_2 \equiv e \bmod (p_2 - 1), \text{ and}$$

$$e_3 \equiv e \bmod (p_3 - 1).$$

The paragraph beginning at col. 6, line 65:

In generalized form, the ciphertext C (i.e., [decrypted] encrypted message M) can be obtained by [the same summation] a recursive scheme as identified above to obtain the ciphertext C from its contiguous constituent sub-tasks C_i.

The paragraph beginning at col. 7, line 1:

Preferably, the recursive CRT method described above is used to obtain either the ciphertext C or the deciphered plaintext (message) M due to its speed. However, there

may be implementations when it is beneficial to use a non-recursive technique in which case the following relationships are used:

$$M \equiv \sum_{i=1}^{t} M_i \, (w_i^{-1} \, (\mathrm{mod}\, p_i)) \cdot w_i \, (\mathrm{mod}\, n)$$

where

$$w_i = \prod_{j \neq i} p_j, \text{ and}$$

k is the number (3 or more) of distinct primes chosen to develop the product n.

The paragraph beginning at col. 7, line 17:

Thus, for example above (k=3), M is constructed from the returned sub-task values M_1, M_2, M_3 by the relationship

$$M \equiv M_1 \, (w_1^{-1} \, (\mathrm{mod}\, p_1)) \cdot w_1 \, (\mathrm{mod}\, n)$$

$$+ M_2 \, (w_2^{-1} \, (\mathrm{mod}\, p_2)) \cdot w_2 \, (\mathrm{mod}\, n)$$

$$+ M_3 \, (w_3^{-1} \, (\mathrm{mod}\, p_3)) \cdot w_3 \, (\mathrm{mod}\, n)$$

where

$$w_1 = p_2 \, p_3, \; w_2 = p_1 \, p_3, \text{ and } w_3 = p_1 \, p_2.$$

The paragraph beginning at col. 7, line 52:

The I/O bus 30 communicatively connects the CPU to a number of exponentiator elements 32a, 32b and 32c. Shown here are three exponentiator elements, although as illustrated by the "other" exponentiators 32n, additional exponentiator elements can be added. Each exponentiator element is a state machine controlled arithmetic circuit structured specifically to implement the relationship described above. Thus, for example, the exponentiator 32a would be provided the values M_1, e_1, and p_1 to develop C_1. Similarly, the exponentiator circuits 32b and 32c develop C_2 and C_3 from corresponding subtask values M_2, e_2, p_2, M_3, e_3, and p_3.

The paragraph beginning at col. 8, line 1:

In order to ensure a secure environment, it is preferable that the cryptosystem 10 meet the Federal Information Processing Standard (FIPS) 140-1 level 3. Accordingly, the elements that make up the CPU 14 would be implemented in a design that will be secure from external probing of the circuit. However, information communicated on the I/O bus 30 between the CPU 14 and the exponentiator circuits 32 (and external memory 34--if present) is exposed. Consequently, to maintain the security of that information, it is first encrypted by the DES unit 24 before it is placed on the I/O bus 30 by the CPU 14. The exponentiator circuits 32, as well as the external memory 34, will also include similar DES units to decrypt information received from the CPU, and later to encrypt information returned to the CPU 14.

The paragraph beginning at col. 8, line 52:

In similar fashion, information is conveyed to or retrieved from the exponentiators 32 by the processor 20 by write or read operations at addresses within the address range 44. Consequently, writes to the exponentiators 32 will use the DES unit 24 to encrypt the information. When that (encrypted) information is received by the exponentiators 32, it is decrypted by on-board DES units (of each exponentiator 32). The result of the task performed by the exponentiator 32 is then encrypted by the exponentiator's on-board DES unit, retrieved by the processor 20 in encrypted form and then decrypted by the DES unit 24.

The paragraph beginning at col. 9, line 24:

Assume, for the purpose of the remainder of this discussion, that the encryption/decryption tasks performed by the cryptosystem 10, using the present invention, employs only three distinct primes, p_1, p_2, p_3. The processor 20 will develop the sub tasks identified above, using M, e, p_1 p_2, p_3 Thus, for example, if the exponentiator 32a were assigned the sub-task of developing C_1, the processor would

develop the values M_I and e_I and deliver (write) these values, with p_I, to the exponentiator 32a. Similar values will be developed by the processor 20 for the sub-tasks that will be delivered to the exponentiators 32b and 32c.

The paragraph beginning at col. 10, line 15:

Alternatively, the host-system 50 may desire to deliver, via the communication medium 60, an encrypted communication to one of the stations 64. If the communication is to be encrypted by the DES scheme, with the DES key encrypted by the RSA scheme, the host system would encrypt the communication, forward the DES key to one of the cryptosystems 10 for encryption via the RSA scheme. When the encrypted DES key is received back from the cryptosystem 10, the host system can then deliver to one or more of the stations 64 the encrypted message.

The paragraph beginning at col. 10, line 25:

Of course, the host system 50 and the stations 64 will be using the RSA scheme of public key encryption/decryption. Encrypted communications from the stations 64 to the host system 50 require that the stations 64 have access to the public key E=(e, n) while the host system maintains the private key D=(d, n) and the constituent primes, $p_1, p_2, \ldots, p_k$). Conversely, for secure communication from the host system 50 to one or more of the stations 64, the host system would retain a public key E' for each station 64, while the stations retain the corresponding private keys D'.

The paragraph beginning at col. 10, line 35:

Other techniques for encrypting the communication could used. For example, the communication could be entirely encrypted by the RSA scheme. If, however, the message to be communicated is represented by a numerical value greater than n-1, it will need to be broken up into blocks size M where

$$0 \leq M \leq n\text{-}1.$$